

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV MATEMATIKY  
FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF MATHEMATICS

# MATEMATICKÉ MODELOVÁNÍ POMOCÍ L-SYSTÉMŮ

MATHEMATICAL MODELLING WITH L-SYSTEMS

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

RŮŽENA JANOUTOVÁ

VEDOUCÍ PRÁCE  
SUPERVISOR

doc. PaedDr. DALIBOR MARTIŠEK,  
Ph.D.

BRNO 2009

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky

Akademický rok: 2009/2010

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

student(ka): Růžena Janoutová

který/která studuje v **bakalářském studijním programu**

obor: **Matematické inženýrství (3901R021)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Matematické modelování pomocí L-systémů**

v anglickém jazyce:

### **Mathematical modelling with L-systems**

Stručná charakteristika problematiky úkolu:

Úkolem práce bude shrnout základní poznatky o Lindenmayerových systémech a demonstrovat některá jejich použití (počítačová grafika, teorie grafů apod).

Cíle bakalářské práce:

Vytvořená práce bude obsahovat základní charakteristiku L-systémů, popis základních programovacích technik k jejich softwarové realizaci a některé možnosti použití. Její součástí bude softwarové řešení.

## **ABSTRAKT**

Tato práce pojednává o L-sytémech a jejich praktickém využití. Práci můžeme pomyslně rozdělit na dvě části – teoretickou a aplikovanou. V teoretické části, tj. v první kapitole, se zabýváme samotnými L-systémy. Nachází se zde jejich zavedení a klasifikace. Aplikovaná část zahrnuje druhou a třetí kapitolu. Druhá kapitola obsahuje přehled využití L-systémů a třetí kapitola je věnována programu, který je součástí bakalářské práce, jeho popisu a návodu k použití.

## **KLÍČOVÁ SLOVA**

L-systémy, abeceda, klasifikace L-systémů, využití L-systémů, popis modelovacího programu

## **ABSTRACT**

This work deals with L-systems and their practical utilization. The work can be divided into two parts – theoretical and applied. In the theoretical part, i.e. in the first chapter, the L-systems are discussed. There can be found their introduction and classification. Applied part includes second and third chapter. Second chapter contains review of L-system's utilization and third chapter is dedicated to program attached to this work, his description and manual.

## **KEYWORDS**

L-systems, alphabet, classification of L-systems, L-systems utilization, simulation program description

JANOUTOVÁ, R. *Matematické modelování pomocí L-systémů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 30 s. Vedoucí bakalářské práce doc. PaedDr. Dalibor Martišek, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Matematické modelování pomocí L-systémů“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno .....

.....

(podpis autora)

Děkuji svému školiteli doc. PaedDr. Daliboru Martiškovi, Ph.D. za odborné konzultace a vedení mé bakalářské práce.

# OBSAH

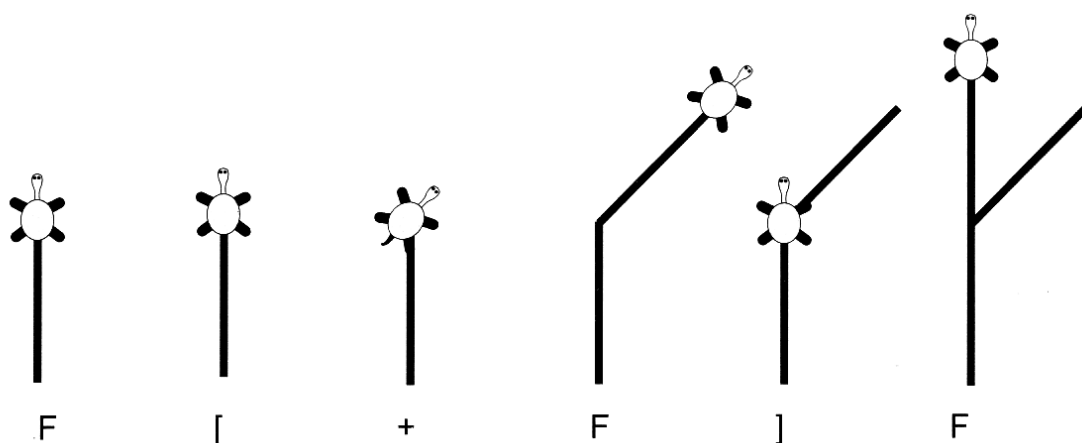
<b>Úvod</b>	<b>6</b>
<b>1 L-systémy</b>	<b>7</b>
1.1 Úvod do L-systémů . . . . .	7
1.2 Klasifikace . . . . .	9
1.2.1 Základní charakteristiky . . . . .	9
1.2.2 Parametrické L-systémy . . . . .	12
1.2.3 Otevřené L-systémy . . . . .	13
1.2.4 Stochastické L-systémy . . . . .	14
<b>2 Využití L-systémů</b>	<b>17</b>
<b>3 Program</b>	<b>20</b>
3.1 Úvod . . . . .	20
3.2 Návod . . . . .	20
3.2.1 Po spuštění . . . . .	20
3.2.2 Soubor . . . . .	22
3.2.3 Nastavení . . . . .	22
3.2.4 O programu . . . . .	24
3.3 Funkce programu . . . . .	25
<b>Závěr</b>	<b>26</b>
<b>Literatura</b>	<b>30</b>

# ÚVOD

Princip L-systémů objevil maďarský biolog Aristid Lindenmayer díky své myšlence, že v oplodněném vajíčku se nachází program, který řídí vývoj organismu. Dnes tento přístup známe jako *paralelní přepisující se systém* využívaný na počítačové simulování růstu živých organismů. Na L-systémy můžeme pohlížet jako na soubor pravidel pro přepisování řetězců, které se po určitém počtu přepsání geometricky interpretují. L-systémy jsou ve své jednodušší formě nazývány jako želví grafika nebo také LOGO, jež bylo základem mnoha jiných programovacích jazyků. Je tomu tak kvůli posloupnosti symbolů, které lze chápat jako příkazy pro pohyb želvy. Nejjednodušší příkazy jsou pohyb dopředu o daný krok, otočení o daný úhel doleva či doprava. Pro složitější simulace se využívají L-systémy s daleko větším množstvím příkazů a dalším rozšířením.

Tato práce se zabývá L-systémy z různých hledisek. Nejdříve uvedeme základní pojmy problematiky L-systémů, dále pak rozdělení, vlastnosti a některé příklady jednotlivých skupin L-systémů. V této práci jsou také uvedeny konkrétněji některé druhy nejvyužívanějších L-systémů. Zjistíme jaké jsou možnosti, co se týče vlastností výstupu apod., a seznámíme se s různými příklady využití L-systémů.

V poslední kapitole se nachází návod k použití programu, který je součástí práce. Popisuje jak používat jednotlivé komponenty uživatelského rozhraní, co lze v programu nastavit i odkazy k příslušným kapitolám v teoretické části. Specifikuje také, pro jaké L-systémy je program schopen zpracovávat data a vykreslovat.



Obr. 1: Pohyb želvy

# 1 L-SYSTÉMY

## 1.1 Úvod do L-systémů

L-systémy se zabývají přepisováním řetězců, co se týče symbolů, ale nás budou nejvíce zajímat převážně grafické výstupy, nejen symbolické. Podle své složitosti se označují různými písmeny, čísly nebo přívlasky. Pro zavedení využijeme nejjednodušší formu.

**Definice 1.1.** 0L-systémem rozumíme každou uspořádanou trojici  $\mathcal{L} = (V, P, \sigma)$ , kde  $V$  značí konečnou množinu znaků, kterou nazýváme *abecedou*,  $P$  je množina přepisovacích pravidel tvaru  $p : a \rightarrow \alpha$ ;  $a \in V$ ;  $\alpha \in V^*$ ;  $p \in P$  a  $\sigma \in V^+$  je axiom, kde  $V^+$  označuje množinu všech konečných neprázdných posloupností utvořených z prvků množiny  $V$ ,  $e$  označuje prázdnou posloupnost a definujeme  $V^* = V^+ \cup \{e\}$ . Množina  $P$  je úplná, tzn. že  $\forall a \in V$  existuje alespoň jedno slovo  $\alpha \in V^*$  takové, že  $a \rightarrow \alpha \in P$ .

**Pozn.:**  $\sigma$  můžeme chápat jako nultou iteraci systému. Prvek  $e$ , prázdnou posloupnost, můžeme v systému využít pro mazání symbolů. Když přepisovací pravidlo bude mít tvar  $p : a \rightarrow e$ , pak symbol náležící tomuto pravidlu bude smazán (nahrazen prázdnou posloupností).

Jak už jsme zmínili v úvodu, symboly můžeme rozumět příkazy pro želvu. Uvedeme pár základních příkazů. Značení zavedeme stejné jaké se nachází v programu. Začneme základními příkazy, což jsou krok dopředu a natočení, tyto prvky náleží množině  $V$  z definice. Krok vpřed značíme písmenem  $F$ . Natočení o daný úhel do kladného směru  $+$  a do záporného směru  $-$ .

**Příklad 1.1.** Zvolíme množinu  $V$ , která obsahuje prvky  $\{F, +, -\}$ , kde  $F$  odpovídá v grafickém výstupu posunutí dopředu s vykreslováním stopy,  $+$  je chápáno jako otočení o úhel  $60^\circ$  do kladného směru (po směru hodinových ručiček) a  $-$  jako otočení do záporného směru.  $P$  obsahuje pouze jedno pravidlo, a to je charakterizováno předpisem  $p : F \rightarrow F - F + +F - F$ , což znamená, že v každé iteraci znak  $F$  nahradíme řetězcem  $F - F + +F - F$ .  $\sigma$  je zadána znakem  $F$ . Zvolíme 4 iterace. Zadání tedy vypadá takto:

L-systém:  $\mathcal{L} = (V, P, \sigma)$ ;

$V = \{F, +, -\}$ ;

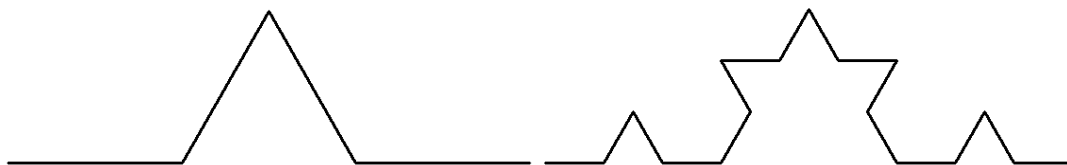
$P = p$ , kde  $p : F \rightarrow F - F + +F - F$ ;

$\sigma = F$ ;

počet iterací = 4, úhel =  $60^\circ$ ;



Tímto L-systémem vykreslenou křivku nazýváme Kochovu podle švédského matematika Nielse Fabiana Helge von Kocha. V programu jsme vygenerovali všechny iterace ze zadání viz. obrázky 1.1 a 1.2.



Obr. 1.1: Kochova křivka: 1. a 2. iterace



Obr. 1.2: Kochova křivka: 3. a 4. iterace

Další rozšíření přináší příkazy pro krok zpět  $B$  a krok bez kreslení  $M$  ( $N$  v případě kroku zpět bez kreslení). Jako další možnost se vyskytne větvení systémů. Želvě přidáme zásobník, kam si bude ukládat souřadnice pozic a úhlů na těchto místech, kam se bude vracet. Do zásobníku se na poslední místo zapíše pozice a úhel ve chvíli, kdy želva narazí na symbol  $[$ , vrátí se na poslední uloženou pozici a načte úhel ve chvíli, kdy v příkazu přečte symbol  $]$ . Dalšími příkazy pro želvu mohou být symboly  $@$ ,  $|$ ,  $C$ , kde symbolem  $@$  určujeme délku kroku želvy. Když želva načte symbol upraví následující krok vynásobením hodnotou následující po symbolu. Např. narazí na  $@0.5$ , délka dalšího kroku bude poloviční oproti původní. Úpravu kroku zavedeme na celý úsek v zásobníku tak, že před uložením pozice do zásobníku určíme úpravu a do konce načtení poslední pozice se bude používat tato délka. Použití tohoto znaku je vidět na obrázku 1.3. Když želva narazí na znak  $|$ , vykoná otočení o  $180^\circ$  nehlédě na nastavený úhel. V tomto případě se dostáváme v L-systémech k zacházení s barvami. Při dosažení symbolu  $C$  želva podle následující hodnoty vybere barvu, kterou bude kreslit, dokud nedojde na další změnu. Barvy by měly být přiděleny konkrétním číslům nebo pomocí parametrizace. Následující tabulka obsahuje přehled příkazů pro želvu. Většina z nich je využita v programu pod stejným označením.

Symbols  $@$  a  $C$  můžeme zahrnout do parametrů kroku, jak je popsáno v kapitole 1.2.2. V případě, že bychom chtěli parametrické vlastnosti i těchto schopností

Symbol	Význam symbolu
F	posun želvy dopředu s kreslením úsečky
M	posun želvy dopředu bez kreslení úsečky
B	posun želvy dozadu s kreslením úsečky
N	posun želvy dozadu bez kreslení úsečky
-	natočení želvy doleva o předem známý úhel
+	natočení želvy doprava o předem známý úhel
[	uložení stavu želvy na zásobník
]	vyjmutí stavu želvy ze zásobníku
@	změna kroku želvy o zadanou hodnotu
	otočení želvy o 180°
C	nastavení barvy kterou želva kreslí

Tab. 1.1: Přehled základních příkazů

želvy, není problém je začlenit mezi neterminální symboly. Mezi další nastavení může patřit tloušťka čáry, vykreslování listů na koncích tzv. větví apod. Nastavení by se provádělo podobně jako u délky kroku. Zavedl by se symbol následovaný hodnotou, o kterou by se změnila daná vlastnost.

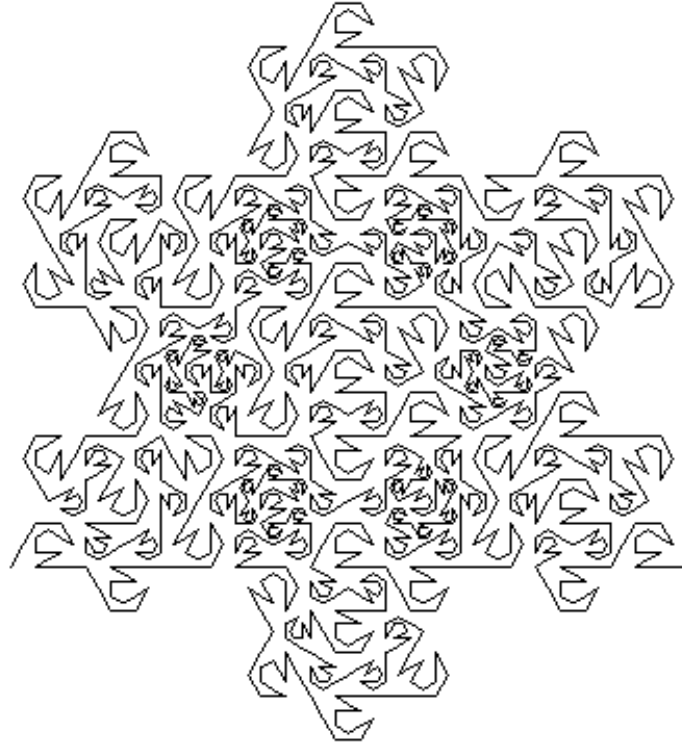
## 1.2 Klasifikace

Definice a označování L-systémů jsou dosti nejednoznačné. Každý autor používá vlastní. Některé z nich jsou však přijímány obecně.

### 1.2.1 Základní charakteristiky

#### 0L,IL-systémy

Interakci chápeme jako ovlivňování funkce symbolů ležících vedle sebe. L-systémy s interakcí značíme IL a bez interakce 0L. Funguje tak, že pro daný symbol existuje předpis (či více předpisů), který určuje podmínky pro vypsání tohoto symbolu. Když tyto podmínky nejsou splněny, symbol se nevypíše nebo se pro něj může určit jiná funkce, to záleží na uživateli případně na programátorovi. Tyto podmínky zavedeme jako levý a pravý kontext. Přepis je realizován ve chvíli, kdy se před symbolem nachází posloupnost znaků odpovídající levému kontextu a za symbolem pravému. Pro interakční nebo také někdy nazývané kontextové L-systémy zavedeme jiné značení prepisovacích pravidel.



Obr. 1.3: Ukázka využití znaku @

**Definice 1.2.** Přepisovací pravidlo  $p \in P$  tvaru  $p : lc < a > rc \rightarrow \alpha (a \in V, \alpha \in V^*)$ , kde  $lc$  je levý kontext a  $rc$  pravý kontext,  $lc, rc \in V^*$  a symbol  $a$  bude přepsán pouze, když stojí uprostřed kontextu.

**Příklad 1.2.** V tomto příkladu ukážeme využití IL-systému k přenášení signálu (symbolu) v řetězci:

L-systém:  $\mathcal{L} = (V, P, \sigma)$ ;

$V = \{a, b\}$ ;

$P = \{p_1, p_2\}$ , kde  $p_1 : b < a \rightarrow b$ ;  $p_2 : b \rightarrow a$ ;

$\sigma = baaaaa$ ;

Vypíšeme prvních pár iterací včetně  $\sigma$  :

*baaaaa*

*abaaaa*

*aabaaa*

*aaabaa*

*aaaaba*

### D0L-systémy

D: odvozené od přívlastku deterministický tzn. mezi přepisovacími pravidly se nenachází žádná dvě pravidla se stejnou levou stranou. Příkladem takového L-systému je např. Kochova křivka uvedená v příkladě 1.1 Opakem jsou L-systémy s náhodným krokem, o nichž se zmíníme v samostatné kapitole 1.2.4.

### P0L-systémy

P: označuje rozpínavé systémy (z anglického propagating). U těchto systémů nenalezneme žádné pravidlo s prázdnou pravou stranou, tzn. že neexistuje symbol, který by se po proběhnutí iterace smazal.

### E0L-systémy

E: (z anglického extended) označujeme rozšířené systémy. Rozšíření se týká *abecedy*  $V$  o terminály. Symboly, jež se v dalších iteracích už nepřepisují, zavedeme jako terminály a jako neterminály ty, které se přepisují.

**Definice 1.3.** E0L-systémem rozumíme každou uspořádanou čtveřici  $\mathcal{L} = (V, P, \sigma, \Sigma)$ , kde  $\Sigma \subseteq V^+$  značí množinu terminálů.

### PT0L-systémy

T: označuje tabulkové systémy. Místo jediného souboru  $P$  přepisovacích pravidel můžeme mít i více takových souborů  $P_1, \dots, P_k$ . Při každém kroku přepisování se smějí aplikovat pouze ta přepisovací pravidla patřící do téhož souboru.

**Příklad 1.3.** Tabulkový L-systém může vypadat následovně.

L-systém:  $\mathcal{L} = (V, P, \sigma)$ ;

$V = \{a, b, p, q\}$ ;

$\sigma = pq$ ;

$P$  je množina tří tabulek přepisovacích pravidel:

$$0 \left\{ \begin{array}{l} p \rightarrow apa \\ q \rightarrow aq \\ a \rightarrow a \\ b \rightarrow b \end{array} \right. \quad 1 \left\{ \begin{array}{l} p \rightarrow bpb, \\ q \rightarrow qb, \\ a \rightarrow a, \\ b \rightarrow b, \end{array} \right. \quad 2 \left\{ \begin{array}{l} p \rightarrow bb, \\ q \rightarrow b, \\ a \rightarrow b, \\ b \rightarrow a, \end{array} \right.$$

Tento systém převádí např. slovo 0112 takto:

$$\begin{aligned}
pq &\xRightarrow{0} apaaq \\
apaaq &\xRightarrow{1} abpbbaaqb \\
abpbbaaqb &\xRightarrow{1} abbpbbaaqbb \\
abbpbbaaqbb &\xRightarrow{2} baabbaabbbbaa
\end{aligned}$$

### 1.2.2 Parametrické L-systémy

Zatím jsme zmínili L-systémy vykreslující jen úsečky o předepsaných délkách, které svírají různé úhly, ale často je třeba, aby se měnila délka, barva, obrazec (nejen úsečka) atp. Vyřešíme to zavedením modulů. Moduly jsou písmena abecedy  $V$  rozšířená o parametry. Modul má syntaktickou podobu  $A(x_1, \dots, x_k)$ . Množina parametrů modulu může být prázdná, ale musí být konečná. Modul má ve své specifikaci formální parametry, které nabývají hodnot skutečných parametrů z množiny reálných čísel. Z parametrů lze tvořit aritmetické a logické výrazy.

**Definice 1.4.** Parametrickým L-systémem rozumíme uspořádaná čtveřice  $\mathcal{L} = (V, P, \sigma, X)$ , kde  $X$  značí množinu formálních parametrů. Pravidla  $p \in P$  pak mají tvar:

$$id : lc < pred > rc : cond \rightarrow succ : moc,$$

kde

- $id$  je označení pravidla,
- $lc, rc$  je levý resp. pravý kontext,
- $pred$  je přepisovaný znak,
- $cond$  je logický výraz nabývající hodnoty 0 nebo 1,
- $succ$  je pravá strana pravidla a
- $moc$  je mocnost pravidla (více v kapitole 1.2.4).

**Příklad 1.4.** Jelikož se přepisovací pravidlo značně rozrostlo, uvedeme příklad, jak by mohla vypadat některá přepisovací pravidla a první iterace, jež jich využije.

L-systém:  $\mathcal{L} = (V, P, \sigma, X)$ ;

$X = \{x\}$ ;

$V = \{A(x), B(x)\}$ ;

$\sigma : A(1)B(3)A(5)$ ;

$P = \{p_1, p_2, p_3\}$ ;

$p_1 : A(x) \rightarrow A(x+1) : 0, 4$ ;

$p_2 : A(x) \rightarrow B(x - 1) : 0, 6;$

$p_3 : A(x) < B(y) > A(z) : y < 4 \rightarrow B(x + z) [A(y)];$

U pravidel  $p_1$  a  $p_2$  máme ukázkou náhodnosti, o které se více dozvíme v kapitole 1.2.4. Tato dvě pravidla modul  $A(x)$  přepíše s pravděpodobností 40 % na modul  $A(x + 1)$  a s pravděpodobností 60 % na modul  $B(x - 1)$ . Třetí pravidlo se přepíše na modulovou posloupnost  $B(x + z) [A(y)]$  za daných kontextových podmínek tj., když modul  $B(y)$  bude mít z levé strany modul  $A(x)$  a z pravé modul  $A(z)$ , a také pokud bude splněna podmínka  $y < 4$ .

První iterace pak může vypadat takto:

$$A(1)B(3)A(5) \Rightarrow A(2)B(6)[A(3)]A(4).$$

Želva při čtení symbolů dokáže rozpoznávat a geometricky interpretovat šířku a barvu čáry, délku kroku nebo úhel natočení atd.

### 1.2.3 Otevřené L-systémy

Otevřenost systému spočívá ve vzájemné komunikaci s okolím. Systém dokáže v každé iteraci informovat okolí a přijímat informace od okolí a přizpůsobovat se jim. Tato vlastnost se velice osvědčuje při simulaci rostlin díky věrohodnosti daného modelu. Např. simulované rostlině díky této vlastnosti můžeme popsat její okolí. Směrem od okolí lze posílat informace o detekci kolizí s překážkami nebo o množství dopadajícího světla, o kyselosti půdy či o přítomnosti hmyzu. Směrem do okolí může rostlina informovat o svém rozložení v prostoru, o množství látek, které z ní vycházejí atp. Tímto dosáhneme při simulaci adaptace s prostředím a přirozenějšího vzhledu rostliny v daném prostředí a poznáme i její vliv na okolí.

Otevřené systémy jsou stochastické (více v kapitole 1.2.4), kontextové, parametrické a rozšířené o komunikační moduly ve tvaru  $?E(x_1, x_2, \dots, x_k)$ . Tento modul slouží k přenosu informací mezi posloupnostmi modulů, tj. rostlinou a jejím okolím. Modul je jako jiný znak vložen do posloupnosti generované L-systémem. Komunikační modul je však na rozdíl od ostatních vložen bez parametrů. Hodnoty parametrů u komunikačního modulu se získávají až v mezikroku mezi každými dvěma iteracemi. Před každým vykonáním iterace projdeme vygenerovaný L-systém a když narazíme na komunikační modul, tak se naplní hodnoty parametrů komunikačního modulu. Poté se pokračuje normálně. Proces se opakuje do vykonání zadaného počtu iterací anebo dokud L-systém má aspoň jeden neterminální znak.

**Příklad 1.5.** Pro názornost uvedeme, jak by mohl takový L-systém vypadat.

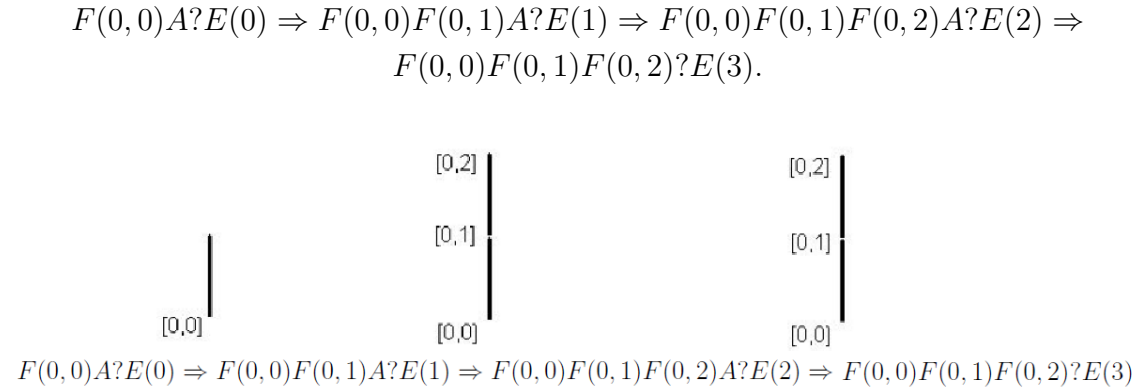
L-systém:  $\mathcal{L} = (V, P, \sigma, X);$

$X = \{x, y\};$

$V = \{F(x, y), A, E(y)\};$

$\sigma : F(0,0)A?E(0);$   
 $P = \{p_1, p_2\};$   
 $p_1 : A >?E(y) : y < 2 \rightarrow F(x, y + 1)A;$   
 $p_2 : A >?E(y) : y > 2 \rightarrow e;$

Systém nám umožňuje ovlivnit objekt tak, aby dosahoval jen do výšky 2. Modul  $F(x, y)$  má význam posunu do bodu  $[x, y]$  z bodu, v němž jsme byli (uvažujeme rovinu). Pravidlo  $p_1$  se aplikuje, pokud není dosaženo výšky 2. V případě dosažení této výšky, se aplikuje pravidlo  $p_2$ , a to smaže posun, který by následoval. Iterace pak budou vypadat následovně



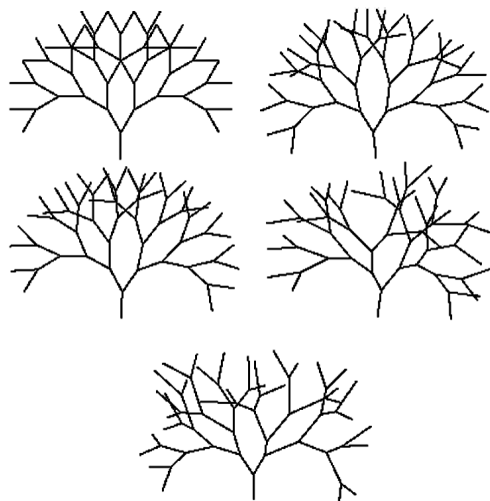
Obr. 1.4: Vykreslení otevřeného systému podle iterací

### 1.2.4 Stochastické L-systémy

vznikají zavedením náhodnosti. Jelikož L-systémy se používají k modelaci biologických struktur hlavně rostlin, stromů, keřů atd., vyskytl se problém pravidelnosti modelů, jež se v přírodě nevyskytuje. Začaly se zavádět různé náhodné prvky (ty však nemění celkový topologický tvar objektu) a jejich obsazení v systémech. Jedná se například o takzvané mutace či simulaci vlivu gravitace při konstrukci 3D modelu. Určíme si pro začátek jen pár nejpoužívanějších náhodných prvků:

- náhodný krok,
- náhodné natočení,
- mutace,
- simulace gravitace.

Nedeterministické L-systémy jsou výsledkem zavedení náhodného kroku. Vybraným symbolům zavedeme více prepisovacích pravidel. Když dojde na přepsání



Obr. 1.5: Následky náhodného kroku a natočení u stejného L-systému

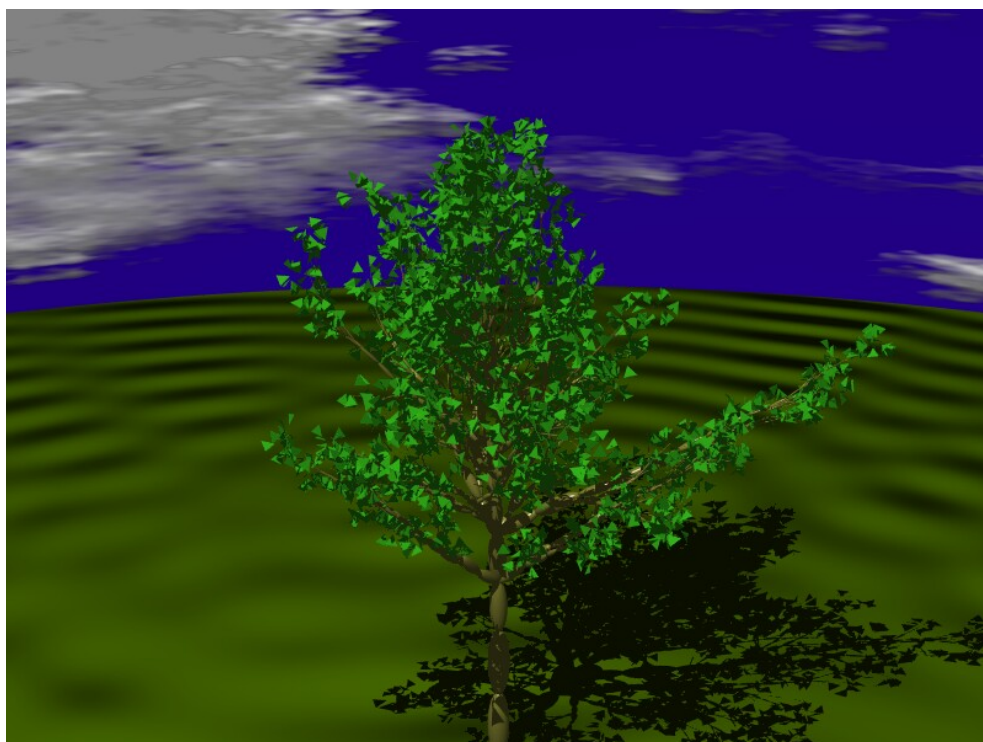
tohoto symbolu, tak se pomocí náhody vybere jedno z pravidel. Výběr můžeme ovlivnit zadáním *mocnosti* každému pravidlu zvlášť. Aplikovat lze určité mnohé další metody na výběr z množiny přepisovacích pravidel pro daný symbol, ale my budeme pracovat s tímto.

Na stejném principu funguje náhodné natočení. Zvolíme dolní a horní mez úhlu, čímž ohraničíme rozsah, z kterého budeme náhodně vybírat úhel. Při každém natočení se bude volit z rozsahu. Podobně lze nastavit mocnosti k daným intervalům úhlů nebo aplikovat nějaké rozdělení na rozsah. Kdybychom chtěli zacházet do detailů, můžeme ještě rozdělit úhly na náhodné a stálé pomocí indexů nebo zavedením nové symboliky.

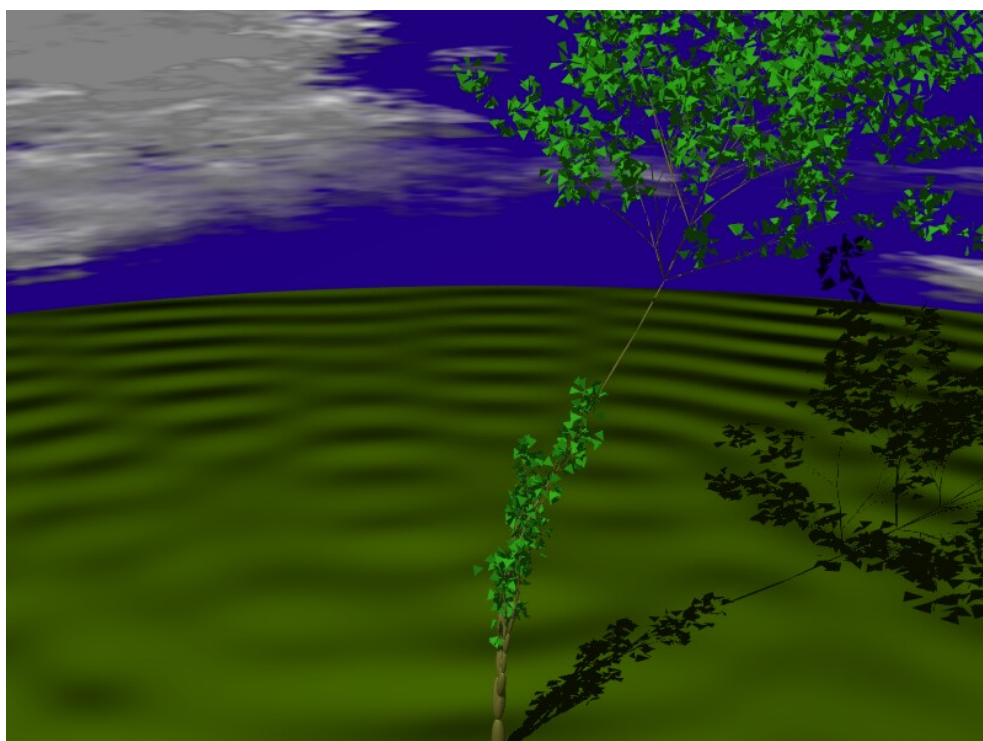
Mutací řešíme problém tvorby skupiny objektů, kde každý objekt je tvořen tímtéž L-systémem. Aby se utvořil efekt přirozenosti, vkládá se do těchto L-systémů prvek mutace. Úroveň mutace nastavujeme podle toho, kolikrát má dojít k mutaci. Mutace samotná spočívá v náhodném vkládání nebo přepisování symbolů řetězce před vykreslením. Do řetězce se vkládají pouze symboly, které v něm už jsou obsaženy.

Abychom nastavili L-systém na simulaci gravitace, museli bychom želvu zorientovat při vytváření každé větve vůči globálnímu souřadnému systému, provést mírné natočení směrem k zemi a posléze provést zpětnou transformaci a pokračovat v pohybu.





Obr. 1.6: Následky mutace stejného L-systému jako 1.7 generované v L-parseru



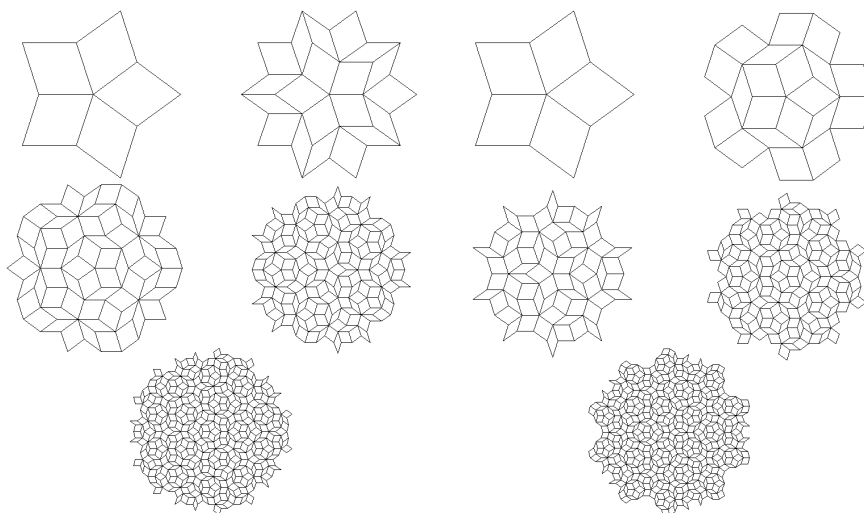
Obr. 1.7: Následky mutace stejného L-systému jako 1.6 generované v L-parseru

## 2 VYUŽITÍ L-SYSTÉMŮ

Hlavní využití L-systémů spočívá v simulaci syntetických organizmů, nejčastěji rostlin. Avšak jejich využití lze najít v mnoha dalších oborech, stručně uvedeme některá z nich.

V oboru formálních jazyků se L-systémy využívají ke generování jazyků. Tabulkové systémy lze používat kromě generování také k překládání z jednoho jazyka do druhého. Jestliže označíme jednotlivé tabulky systému pomocí symbolů z nějaké konečné abecedy  $\Delta$ , potom každý řetězec  $u \in \Delta^*$  můžeme chápat jako předpis určující tabulky a jejich pořadí, v němž se mají při generování slova používat. Předepsaným způsobem vznikají slova jistého jazyka  $L_u \subseteq V^*$ , kde  $V$  je abeceda systému. Daný systém tedy určuje jisté zobrazení  $\Delta^* \rightarrow P(V^*)$ . V případě, že systém je navíc deterministický, určuje zobrazení  $\Delta^* \rightarrow V^*$ .

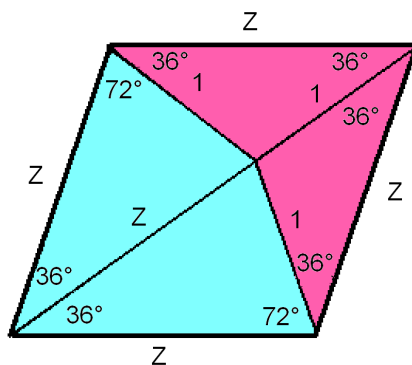
V počítačové grafice, geometrii, umění i architektuře se vyskytuje problém vyplnění určité plochy geometrickými obrazci tzv. dlaždicemi beze zbytku. Vzory vytvořené těmito obrazci se mohou opakovat, vznikají pak periodické mozaiky. Mozaika může být i nepravidelná, skládá-li se z náhodných či nepravidelných tvarů. Mezi těmito druhy mozaik se nachází tzv. aperiodické, jejichž vzory se neopakují, přestože se skládají ze základních tvarů (lze matematicky dokázat, že pro některé pečlivě vybrané základní tvary lze sestavit nekonečné množství možností, kterými tyto tvary vyplňují rovinu). Problematikou periodických a aperiodických mozaik se zabýval i Johannes Kepler. Zejména periodické mozaiky se dají tvořit pomocí L-systémů.



Obr. 2.1: Penroseovy Mozaiky generované L-systémem

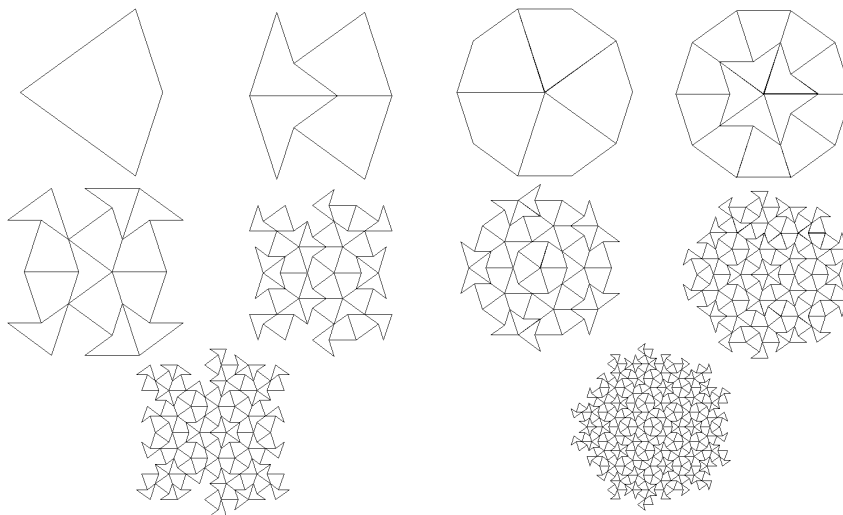
Periodickými a aperiodickými mozaikami se zabýval i vědec Roger Penrose.

Zkoumal základní tvary pomocí nichž se dají vytvářet aperiodické mozaiky. Došel k závěru, že stačí použít pouze dva základní tvary. Získáme je rozdělením kosočtverce s vnitřními úhly  $72^\circ$  a  $108^\circ$  tak, jak to naznačuje obrázek 2.2.  $Z$  představuje hodnotu zlatého řezu, která odpovídá  $\frac{1+\sqrt{5}}{2}$ .



Obr. 2.2: Kites&Darts

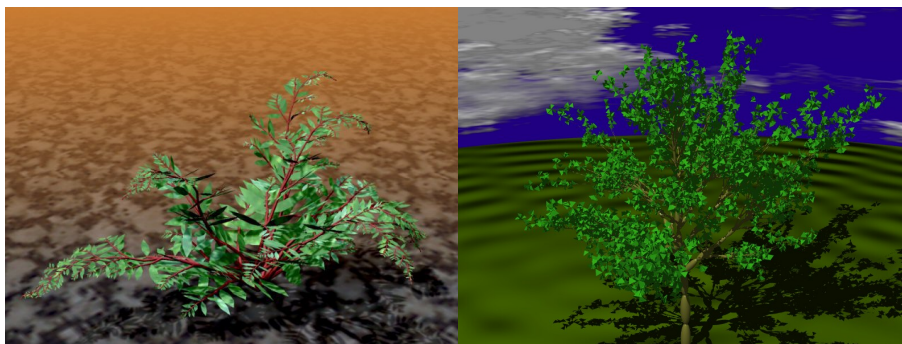
Modrý tvar nazýváme *kite* a fialový *dart*. Díky tomu se mozaiky vytvořené těmito tvary nazývají *Kites and Darts*. Tvar složený ze dvou kites a dvou darts se nazývá *ace*, ten je pak základem pro tvorbu složitějších mozaik.



Obr. 2.3: Kites&Darts

Na vykreslovacím principu L-systémů pracuje program zvaný L-parser (L-System Parser/Mutator), vytvořený Laurencem J. Laprem. Zpracovává především 3D modely, ale dá se nastavit i na tvorbu 2D modelů. Kromě základních příkazů, jako posunutí dopředu podle vektoru, se zde setkáme se stochastickými systémy dokonce

i mutací, simulací gravitace, parametrickými systémy. Lze v něm vytvořit modely jako např. na obrázcích 2.4.



Obr. 2.4: Modelováno v Lparsru

Existuje i mnoho dalších programů využívajících L-systémů. Kromě modelování jednotlivých objektů lze L-systémy využívat na modelování skupiny objektů. Díky mutaci se dá vymodelovat více objektů, které se budou z širšího pohledu jevit stejně, ale v detailech se budou lišit. Mutace se využívá ve filmové grafice při tvorbě lesů, nebe, davu lidí, umělých měst atp. Model vytvořený L-systémem je většinou použit jen jako kostra a pomocí jiných grafických principů je dokreslen zbytek. Kromě filmové grafiky se L-systémy podílejí na simulacích ve vědním oboru tzv. umělý život (artificial life) spolu s jinými metodami. Z této oblasti pochází výraz *biomorf* od Richarda Dawkinse, který chtěl vizualizací biomorfů demonstrovat kreativní potenciál selekčně mutačního mechanismu, používaného v evolučních technikách.

## 3 PROGRAM

### 3.1 Úvod

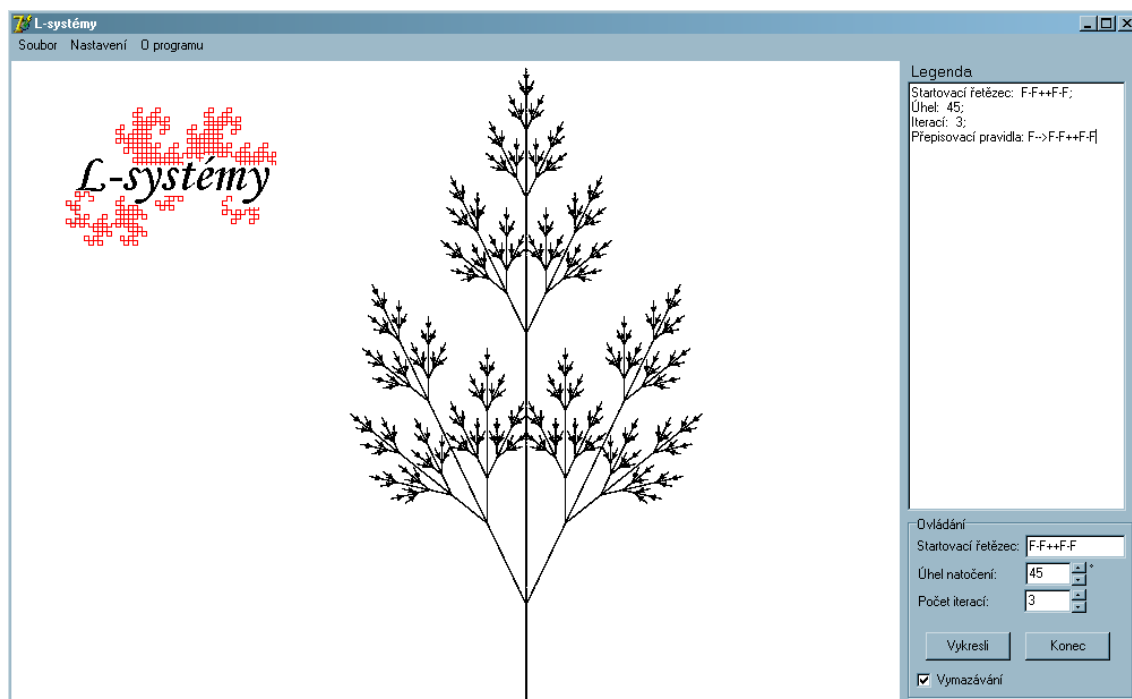
Součástí této práce je program na vykreslování L-systémů, vytvořený v programovacím jazyce Delphi. V předchozích kapitolách jsme viděli některé ukázky vytvořené v tomto programu. Kromě základních 0L-systémů je možno v programu vykreslovat i stochastické systémy a EL-systémy. K dalším funkcím a nastavením se dostaneme podrobněji v dalších kapitolách.

### 3.2 Návod

Kapitola je zaměřena na seznámení uživatele s programem, jeho možnostmi nastavení, ovládání apod.

#### 3.2.1 Po spuštění

Po spuštění se objeví uživatelské rozhraní programu. Obsahuje místo pro vykreslování modelů, legendu, nastavení základních parametrů L-systému a hlavní menu, jak je vidět na obrázku 3.1.



Obr. 3.1: Program po spuštění

## Obraz modelu

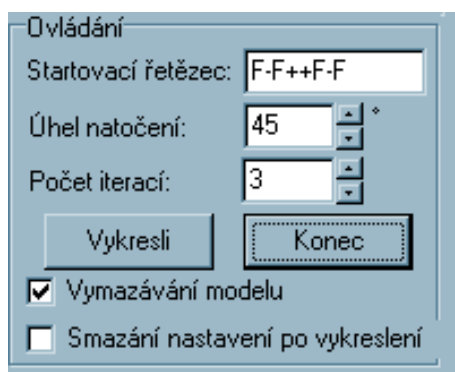
Sekce je určena na vykreslování modelu vytvořeného L-systémem, který jsme nastavili. Na zobrazované části zůstane předchozí model nebo úvodní logo, dokud se nestiskne tlačítko „Vykresli“. Poté se vykreslí model nový. Je umožněno přepisovat stávající model novým, ale i vykreslovat na zcela čistou plochu. Pokud chceme nastavit přepisování modelů, musíme v základním nastavení nechat neoznačenou kolonku „Vymazávání“. Při opačné situaci se každý nový model bude vykreslovat na novou plochu.

## Legenda

V legendě se zobrazují nastavené údaje L-systému. Mezi ně patří startovací řetězec, daný úhel nebo rozmezí úhlů, počet iterací a přepisovací pravidla, neterminální znaky, stav náhodnosti apod. Startovací řetězec odpovídá  $\sigma$  z definice 1.1. Další označení je stejné jako v teoretické části, viz. kapitola 1.1. Legenda se aktualizuje při každém vykreslení.

## Základní nastavení

Zde lze nastavit pouze startovací řetězec, základní úhel a počet iterací. Vše se vpišuje do příslušně označených kolonek nebo se může nastavit pomocí šipek vedle kolonky podle obrázku 3.2. Základní úhel platí pouze, pokud je L-systém nastaven bez náhodnosti. Iterace se počítají bez startovacího řetězce, tzn. že první iterace je přepis startovacího řetězce. Funkci kolonky „Vymazávání“ známe ze sekce Obraz modelu. Dále jsou zde dvě tlačítka. „Vykresli“, jehož činnost jsme uvedli, a „Konec“. Po stisknutí posledně zmíněného tlačítka následuje ukončení aplikace bez uložení dat.



Obr. 3.2: Základní nastavení

## Hlavní menu

Pokud nechceme vykreslovat pouze základní L-systémy, bude nás zajímat hlavní menu. Zde jsou položky „Soubor“, „Nastavení“ a „O programu“. Více k těmto položkám se dozvíme v následujících podkapitolách 3.2.2, 3.2.3 a 3.2.4.

### 3.2.2 Soubor

V této nabídce lze vybrat z možností „Načti“ a „Ulož“. Pokud chceme uložit nastavení, která jsme si zadali, nebo chceme načíst některá nastavení již uložená, využijeme této nabídky.

#### Načti

Po kliknutí na tuto volbu lze vybrat některý z uložených L-systémů. Program sám obsahuje přednastavené L-systémy, z nichž je možno vybírat. Uživatel si může vybrat i své vlastní uložené L-systémy. Program podporuje jen specificky uložené soubory.

#### Ulož

Máme možnost ukládat své vlastní nastavení L-systému. Vytvoří se soubor s daty, který může být později znovu načten. Program umožňuje ukládání samostatných obrázků, vytvořených a vykreslených modelů.

### 3.2.3 Nastavení

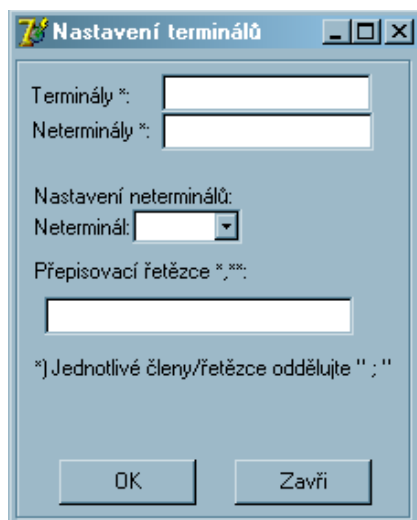
Pokud chceme změnit nastavení, které nelze zadat v základním okně, využijeme této nabídky.

#### Terminály

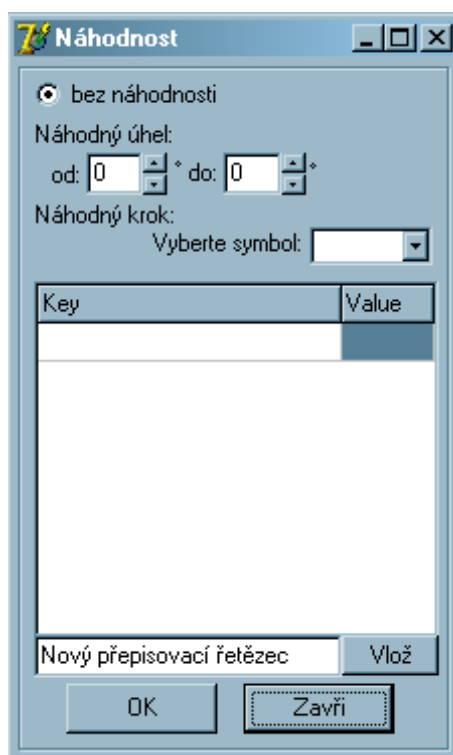
Menu zobrazíme stisknutím volby „Terminály“ v „Nastavení“. Zadávají se terminální a neterminální znaky do označených kolonek. Význam znaků jsme uvedli v kapitole 1.1. Při zadávání oddělujeme jednotlivé symboly čárkou. Po rozlišení neterminálů máme možnost nastavit jejich přepisovací pravidla. Menu se ukončuje dvěma způsoby. Buď tlačítkem „OK“, které data uloží do aktuálního L-systému, nebo pomocí tlačítka „Zavři“, jež menu zavře bez uložení dat.

#### Náhodnost

Jestliže chceme vykreslit stochastický model, musíme nastavit náhodnost menu 3.4, jež se nám objeví po volbě „Nastavení“ a „Náhodnost“. Z kapitoly o stochastických



Obr. 3.3: Menu pro nastavení terminálních a neterminálních symbolů



Obr. 3.4: Menu pro nastavení náhodnosti

L-systémech 1.2.4 zde můžeme nastavit náhodný krok a natočení. Základní nastavení je bez náhodnosti, když necháme tuto volbu neoznačenu bude L-systém stochastický. Náhodný úhel nastavíme pomocí rozsahu. Z něj se volí náhodně (pomocí funkce rand implementované v Delphi). Náhodný krok je realizován pomocí dodatečného



nastavení neterminálních znaků, tzn. připsání dalších prepisovacích pravidel jednotlivým neterminálním znakům, není nutné všem, a nastavení mocnosti ke každému pravidlu. Z prepisovacích pravidel se vybírá podobně jako u náhodného natočení s tím rozdílem, že každá možnost má svoji váhu. Tlačítka „OK“ a „Zavři“ fungují stejně jako v předchozím menu.

## Funkce

Nastavení „Funkce“ souvisí s grafickou interpretací symbolů. Jelikož uživateli je dovoleno používat libovolné znaky, musí zde nastavit jejich funkci. Program má výchozí nastavení funkcí ke znakům dle tabulky 3.1. Přenastavit je lze jednoduše. Ke každé funkci lze zvolit množinu symbolů, které budou mít stejnou funkci. Zadávání symbolů je na stejném principu jako u zadávání terminálních a neterminálních znaků, tzn. oddělování jednotlivých znaků čárkou. Udělování funkcí symbolům probíhá až po dokončení poslední iterace a nebo pokud se v řetězci nacházejí pouze terminální znaky. Obě tlačítka zde fungují stejně jako v předešlých menu.

Symbol	Význam symbolu
F	posun želvy dopředu s kreslením úsečky
M	posun želvy dopředu bez kreslení úsečky
B	posun želvy dozadu s kreslením úsečky
N	posun želvy dozadu bez kreslení úsečky
-	natočení želvy doleva o předem známý úhel
+	natočení želvy doprava o předem známý úhel
[	uložení stavu želvy na zásobník
]	vyjmutí stavu želvy ze zásobníku
	otočení želvy o 180°

Tab. 3.1: Přehled základních příkazů

### 3.2.4 O programu

Zde nalezneme stručný přehled kapitoly 3.2, obdoba „help“ v jiných programech. Uživatel se dozví stručné informace o nastavení L-systému v tomto programu. Dále zde najdeme obecné informace o programu (datum vzniku, autor apod.).

### 3.3 Funkce programu

Prohlédneme si činnost programu. Program je rozdělen na čtyři jednotky: hlavní, grafickou, výpočtovou a datovou. Každá má svou specifickou funkci v programu. Kromě hlavní jednotky obsahuje každá svou třídu, která pak plní všechny požadavky na danou jednotku.

#### Hlavní

Činnost této jednotky je pouze ve zpracovávání uživatelského rozhraní, nastavení funkcí a procedur jednotlivých komponent nebo ukončení celé aplikace.

#### Grafická

Grafická část obstarává vzhled modelu. Do jednotky vstupuje pouze konečný řetězec obsahující pouze znaky, které se budou vykreslovat, tzn. znaky uvedené v „Nastavení“ → „Funkce“. Tato jednotka sama ošetřuje velikost jednoho kroku vzhledem k velikosti vykreslovacího pole a celkového modelu. Vypočítává počátek kreslení tak, aby model byl uprostřed vykreslované části. Obsahuje i další jiné funkce, které se starají o grafický výstup.

#### Výpočtová

Asi nejobsáhlejší jednotka je výpočtová. Obsahuje následující funkce nebo procedury:

- ošetření vstupních hodnot,
- přepisování řetězců,
- náhodné výběry,
- zapisování do aktuálního L-systému,
- zpracování konečného řetězce do vykreslovací formy,
- mazání L-systému z paměti.

Stará se o veškeré výpočty a převody, které jsou potřeba při výpočtech iterací, zapsání dat do a z L-systému a všechna vstupní ošetření.

#### Datová

Veškerá práce při ukládání nastavení nebo modelu z programu nebo načítání dat do programu se realizuje v této jednotce.

## ZÁVĚR

Problematika L-systému je velmi rozsáhlá a tato práce obsahuje jen velmi malý zlomek informací. Snahou bylo přiblížit toto téma alespoň v základním rozsahu. Popsali jsme na jakém principu L-systémy pracují. Uvedli jsme stručně nejpužívanější klasifikaci. Jednu kapitolu jsme věnovali jejich využití, i když by se jejich využití mohla věnovat celá práce. Prvními dvěma kapitolami jsme uvedli L-systémy, abychom poslední kapitolu mohli věnovat modelovacímu programu, který je součástí práce. Po přečtení této práce by mělo být jasné, na jakých principech pracuje přiložený program. Poslední kapitola byla pojata jako manuál k programu a stručně objasňuje jak program pracuje.

Abychom obsáhli téma L-systémy byla by zapotřebí daleko rozsáhlejší práce, proto uvedeme některé možnosti, kterými by se tato práce dala rozšířit. Jedna z možností je věnovat větší pozornost jednotlivým rozdělením, uvést si podrobnější popis, praktické důsledky, využití a více příkladů. Mohli bychom uvést i další méně používané rozdělení. Parametrické a otevřené L-systémy mají vliv na reálnější modelování živých organismů, proto by se rozšíření mohlo týkat hlavně nich. U parametrických L-systémů bychom mohli věnovat pozornost různým parametrům, tzn. jak ovlivňují efekty a kdy je vhodné je použít atd. U otevřených L-systémů můžeme posuzovat vhodné místo v systému na vložení komunikačních modulů nebo jaké parametry a jak často je potřeba hlídat pomocí komunikačních modulů.

Program přiložený k této práci zpracovává jen základní L-systémy bez otevřenosti, bez parametrů a náhodnost využívá jen na základní úrovni náhodného kroku a náhodného natočení. Přesto program obsahuje rozsáhlé procedury na výpočetní algorytmy, grafické výstupy a ošetření vstupů díky velké volnosti v zadávání nastavení L-systému. Program pracuje s ukládáním a načítáním L-systému. Nastavení všech možností je pomocí volby „nastavení“ na hlavním panelu nebo v hlavním okně v dolním pravém rohu. Rozšíření programu by mohlo být uskutečněno obsazením parametrů, komunikačních modulů, mutací a také by mohl být rozšířen o 3D verzi.

# ZÁKLADNÍ POJMY A ZNAČENÍ

Uvedeme si seznam pojmů a používané značení:

$V$	... konečná abeceda
$e$	... prázdná posloupnost množiny $V$
$V^+$	... množina všech konečných neprázdných posloupností prvků $V$
$V^*$	... $V^* = V^+ \cup \{e\}$
$P$	... množina přepisovacích pravidel
$\sigma$	... axiom $\sigma \in V^+$
$\Sigma$	... množina terminálů, $\Sigma \subseteq V^+$
$lc$	... levý kontext, $lc \in V^*$
$rc$	... pravý kontext, $rc \in V^*$
$modul$	... parametrický symbol
$?E(x_1, x_2, \dots, x_k)$	... komunikační modul
$\xRightarrow{a}$	... použití přepisovacího pravidla z tabulky $a$

# SEZNAM TABULEK

1.1	Přehled základních příkazů . . . . .	9
3.1	Přehled základních příkazů . . . . .	24

## SEZNAM OBRÁZKŮ

1	Pohyb želvy . . . . .	6
1.1	Kochova křivka: 1. a 2. iterace . . . . .	8
1.2	Kochova křivka: 3. a 4. iterace . . . . .	8
1.3	Ukázka využití znaku @ . . . . .	10
1.4	Vykreslení otevřeného systému podle iterací . . . . .	14
1.5	Následky náhodného kroku a natočení u stejného L-systému . . . . .	15
1.6	Následky mutace stejného L-systému jako 1.7 generované v L-parseru . . . . .	16
1.7	Následky mutace stejného L-systému jako 1.6 generované v L-parseru . . . . .	16
2.1	Penroseovy Mozaiky generované L-systémem . . . . .	17
2.2	Kites&Darts . . . . .	18
2.3	Kites&Darts . . . . .	18
2.4	Modelováno v Lparsru . . . . .	19
3.1	Program po spuštění . . . . .	20
3.2	Základní nastavení . . . . .	21
3.3	Menu pro nastavení terminálních a neterminálních symbolů . . . . .	23
3.4	Menu pro nastavení náhodnosti . . . . .	23

## LITERATURA

- [1] ZELINKA Ivan, VČELAŘ František, ČANDÍK Marek. *Fraktální geometrie - principy a aplikace*. Praha: BEN-technická literatura, 2006.
- [2] CHYTIL, Michal. *Automaty a gramatiky*. Praha: SNTL, 1984
- [3] ŽÁRA a kol. *Moderní počítačová grafika*. Praha: Grada, 1998
- [4] PRUSINKIEWICZ, Przemyslaw, LINDENMAYER Aristid. *The Algorithmic Beauty of Plants*. Elektronická edice *The Algorithmic Beauty of Plants*. New York: Springer-Verlag, 1990, dotisk v 1996.
- [5] *Fraktály v počítačové grafice*[seriál online]: TIŠNOVSKÝ, Pavel, 2005- . [cit.2010-05-25]. URL:<<http://www.root.cz/serialy/fraktaly-v-pocitacove-grafice/>>